

PENERAPAN METODE GERAKAN OBYEK UNTUK PENGAMBILAN CITRA *DIGITAL* PADA *OBTION REMOTE* VERSI 1.0

Astika Ayuningtyas, Agus Basukesti, NurCahyani Dewi Retnowati

Jurusan Teknik Informatika

Sekolah Tinggi Teknologi Adisutjipto Yogyakarta

informatika@stta.ac.id

ABSTRACT

Development of the world's information and communication technology-based growing rapidly. The development of these technologies has given rise to another dimension in human life, namely the need for computer use. Application of machine vision applications where a computer has the ability to understand the data in the form of image and make decisions based on data obtained from a real object, in this case is generated through the capture image from webcam.

*The images (digital images) of a real object through a webcam can be done in many ways capture the object. In this research object retrieval using an object of activity is captured on the webcam through the motions of objects. Digital image acquisition is performed with the tool in the form of software (*Obtion Remote Version 1.0*). *Obtion* (*Object Detection*) method works with the system of comparison between that time frame the previous frame captured by the camera at the time of recording the object. Comparison of the difference in the frame will be accommodated in a place that will be used to support the execution of an object by counting the number of pixels that exceed the value of the difference in number of field values with regulator frame to frame changes in the location of the object motion*

Obtion Remote Version 1.0 will take a digital image of the object with trigger object motion detection performed. Digital image acquisition (capture the object with a webcam) will be done automatically if the object has been successfully detected. The detected object will be stored automatically in the directory are executed as a digital image storage.

Keywords: Method of Object Motion, Digital Image, Obtion Remote.

ABSTRAK

Perkembangan dunia informasi dan komunikasi berbasis teknologi semakin pesat. Perkembangan teknologi tersebut telah melahirkan dimensi lain dalam sisi kehidupan manusia yaitu penggunaan kebutuhan akan komputer. Aplikasi mesin visi merupakan suatu aplikasi dimana komputer memiliki kemampuan untuk memahami data berupa *image* dan mengambil keputusan berdasarkan data yang didapat dari suatu obyek nyata, dalam hal ini adalah *image* yang dihasilkan melalui *capture* dari *webcam*.

Hasil gambar (citra *digital*) dari suatu obyek nyata melalui *webcam* dapat dilakukan dengan bermacam cara pengambilan obyek. Pada penelitian ini dilakukan pengambilan obyek dengan memanfaatkan aktivitas obyek yang tertangkap pada *webcam* yaitu melalui gerakan obyek. Pengambilan citra *digital* ini dilakukan dengan alat bantu yang berbentuk perangkat lunak (*Obtion Remote* Versi 1.0). Metode *object detection* (*obtion*) bekerja dengan sistem perbandingan antara *frame* saat itu dengan *frame* sebelumnya yang tertangkap kamera pada saat perekaman obyek. Perbandingan nilai selisih pada *frame* tersebut akan ditampung pada sebuah tempat yang nantinya akan digunakan untuk pendukung pengekskusi obyek dengan

menghitung banyaknya piksel yang nilai selisih jumlahnya melebihi nilai bidang *frame* dengan pengatur perubahan letak *frame* terhadap gerakan obyek.

Obtion *Remote* Versi 1.0 akan mengambil citra *digital* obyek dengan pemicu deteksi gerakan yang dilakukan obyek. Pengambilan citra *digital* (*capture* obyek dengan *webcam*) akan dilakukan secara otomatis apabila obyek telah berhasil terdeteksi. Hasil eksekusi obyek yang terdeteksi akan tersimpan otomatis pada direktori yang dieksekusi sebagai tempat penyimpanan citra *digital*.

Kata kunci : Metode Gerakan Obyek, Citra Digital, Obtion Remote.

1. Latar Belakang

Perkembangan dunia informasi dan komunikasi berbasis teknologi semakin pesat. Perkembangan teknologi tersebut telah melahirkan dimensi lain dalam sisi kehidupan manusia yaitu penggunaan kebutuhan akan komputer. Salah satunya pada sisi perangkat lunak, bagian dari perangkat komputer ini menjadi bagian terpenting dalam perkembangan teknologi. Aplikasi mesin visi merupakan suatu aplikasi dimana komputer memiliki kemampuan untuk memahami data berupa *image* dan mengambil keputusan berdasarkan data yang didapat dari suatu obyek nyata, dalam hal ini adalah *image* yang dihasilkan melalui *capture* dari *webcam*.

Mendapatkan hasil gambar (citra *digital*) dari suatu obyek nyata melalui *webcam* dapat dilakukan dengan bermacam cara pengambilan obyek. Cara tersebut disesuaikan dengan tujuan pengambilan terhadap obyek yang ditangkap *webcam*. Pada penelitian ini dilakukan pengambilan obyek dengan memanfaatkan aktivitas obyek yang tertangkap pada *webcam* yaitu melalui gerakan obyek. Pengambilan citra *digital* ini dilakukan dengan alat bantu yang berbentuk perangkat lunak (*software*). *Software* tersebut (Obtion *Remote* Versi 1.0) akan membantu melakukan perekaman dan pendeteksian obyek dengan metode *obtion* (*object detection*). Metode *object detection* (*obtion*) bekerja dengan sistem perbandingan antara *frame* saat itu dengan *frame* sebelumnya yang tertangkap kamera pada saat perekaman obyek. Perbandingan nilai selisih pada *frame* tersebut akan ditampung pada sebuah tempat yang nantinya akan digunakan untuk pendukung pengeksekusian obyek dengan menghitung banyaknya piksel yang nilai selisih jumlahnya melebihi nilai bidang *frame* dengan pengatur perubahan letak *frame* terhadap gerakan obyek.

2. Landasan Teori

Computer Vision

Computer Vision adalah suatu aplikasi dengan metode dimana komputer mempunyai kemampuan untuk memahami data berupa *image* dan fungsi untuk mengolah data dan mengambil keputusan berdasarkan data yang didapat dari suatu obyek nyata, dalam hal ini adalah *image* (gambar) yang dihasilkan melalui *capture* dari *webcam*.

Citra

Citra adalah suatu kemiripan atau *imitasi* dari suatu obyek atau benda. Citra yang terlihat merupakan cahaya yang direfleksikan dari sebuah obyek. Sumber cahaya menerangi obyek, obyek memantulkan kembali sebagian dari berkas cahaya tersebut dan pantulan cahaya ditangkap oleh alat-alat optik, misalnya mata manusia, kamera, scanner, dsb, kemudian direkam. Citra yang tampak dalam matematis merupakan kumpulan nilai-nilai tertentu yang membentuk suatu pola berdasarkan keadaan yang telah dikondisikan. Citra sebagai keluaran dari suatu sistem perekaman data dapat bersifat optik berupa foto, *analog* berupa sinyal video seperti gambar pada

monitor televisi, *digital* yang dapat langsung disimpan pada media penyimpanan magnetik (Idhawati, 2011).

Citra *digital* disimpan dalam format *digital* (dalam bentuk *file*). Hanya citra *digital* yang dapat diolah menggunakan komputer. Jenis citra lain jika ingin diolah menggunakan komputer harus terlebih dahulu citra tersebut diubah menjadi citra *digital*. Citra diskrit atau citra *digital* dihasilkan melalui proses *digitalisasi* terhadap citra kontinu. Contoh : kamera *digital*, *scanner*. Citra digital merupakan fungsi intensitas cahaya $f(x,y)$, dimana harga x dan y merupakan koordinat spasial dan harga fungsi tersebut pada setiap titik (x,y) merupakan tingkat kecermerlangan atau intensitas cahaya citra pada titik tersebut.

Deteksi Gerakan

Deteksi gerakan merupakan operasi dasar yang banyak digunakan dalam aplikasi Mesin Visi. Operasi ini digunakan untuk mendeteksi adanya gerakan dari suatu obyek pada daerah tangkapan kamera. Terdapat beberapa teknik untuk mendeteksi adanya gerakan dalam sebuah *frame*, yang termasuk kompleks adalah menggunakan teknik *Optical Flow*. Teknik *optical flow* tujuannya untuk membedakan antara daerah objek dengan daerah yang kosong, *optical flow* sangat terpengaruh oleh cahaya, biasanya *optical flow* ditandai oleh kotak-kotak kecil pada objek yang terdeteksi.

Deteksi Obyek Terhadap Latar

Deteksi obyek juga merupakan aplikasi dasar mesin visi yang biasanya dilanjutkan dengan operasi pengenalan pola atau klasifikasi. Dengan melakukan deteksi obyek, maka dapat diambil informasi piksel mana dalam *frame* yang ditangkap merupakan bagian dari suatu obyek dan bagian mana yang merupakan latar.

Salah satu cara yang dapat digunakan untuk mendeteksi obyek adalah dengan membandingkan *frame* pada saat itu dengan *frame* yang berisi pemandangan latar tanpa adanya obyek yang hendak dideteksi (alih-alih dibandingkan dengan *frame* sebelumnya).

Piksel

Piksel (*picture element* / unsur gambar) adalah titik–titik kecil. Gambar apapun yang tampak di layar komputer sebenarnya tersusun dari titik-titik kecil yang disebut piksel. Jika beberapa piksel diletakkan berderet, akan tampaklah suatu garis. Jadi semua garis, sehalus apapun tampaknya di layar komputer, sebenarnya adalah deretan piksel. Sebuah piksel memang bisa dianggap sebagai sebuah titik, namun dalam kenyataannya, piksel-piksel lebih mirip dengan persegi panjang kecil yang tingginya tidak sebanding dengan lebarnya.

Kontras

Kontras suatu citra adalah perbedaan antara nilai piksel tergelap dengan piksel yang paling terang dalam citra. Citra dikatakan kontrasnya rendah jika perbedaan kecil, sedangkan kontras tinggi jika perbedaannya besar. Tidak semua perangkat keras pencitraan menghasilkan citra dengan kontras yang baik (biasanya kontras rendah), untuk itu perlu dilakukan operasi peningkatan kontras untuk citra.

Kontras yang optimal adalah jika perbedaan tersebut sesuai dengan kedalaman bit dari format citra. Misal citra keabuan 8 bit memiliki kontras optimal sebesar 255 karena piksel tergelap adalah hitam dengan nilai 0 dan paling terang putih dengan nilai 255.

Operasi kontras merupakan operasi titik (hanya bergantung pada nilai piksel, bukan pada lokasi citra), dengan berbagai persamaan yang dapat diterapkan. Salah satu persamaan yang dapat digunakan adalah sebagai berikut :

$$\begin{aligned} R_o &= Gain_R * (R_i - Center_R) + Center_R \\ G_o &= Gain_G * (G_i - Center_G) + Center_G \\ B_o &= Gain_B * (B_i - Center_B) + Center_B \end{aligned} \dots\dots\dots(1)$$

Dimana *Gain* adalah faktor penguatan kontras sedangkan *Center* adalah pusat penguatan kontras. Kedua koefisien ini dapat seragam untuk semua elemen warna ataupun berbeda.

Komponen Penangkap Frame Kamera

Komponen standar dari *Delphi* tidak dapat digunakan untuk mengakses kamera *digital* atau *webcam*. Oleh karena itu, diperlukan komponen tambahan untuk menangkap (*capture*) citra atau *frame* dari *webcam* baik untuk penangkapan citra secara tunggal maupun dalam bentuk citra dinamik (*multiframe*).

Banyak komponen yang dapat digunakan untuk keperluan itu terdapat di *internet*. Namun, hampir semua komponen tersebut merupakan produk *shareware* yang umumnya memiliki beberapa kekurangan, yaitu :

1. Hanya dapat dipakai dalam waktu tertentu.
2. Dapat digunakan tanpa batasan waktu, tetapi komponen tersebut meninggalkan jejak dengan mencantumkan tulisan tertentu pada setiap *frame* yang ditangkap dari kamera.

Pembatasan waktu dan penambahan tulisan tersebut akan dihilangkan apabila sudah membeli lisensi dari pembuatnya.

Komponen *JHVideoCap* ini merupakan paket komponen yang dibuat oleh J.Huebler dari Jerman. *JHVideoCap* adalah produk yang bersifat *freeware* dan boleh digunakan untuk sembarang aplikasi tanpa harus membayar. Untuk menggunakan komponen *JHVideoCap*, tidak diperlukan lisensi sama sekali.

Paket *JHVideoCap* terdiri atas dua komponen yaitu :

- a. *TVideoCap* yang digunakan untuk menangkap *frame* langsung dari *webcam*.
- b. *TVideoDisp*, yang digunakan untuk menampilkan *frame bitmap*.

Hasil Kuesioner Aktivasi Gerakan di Depan Komputer

Penting untuk mengetahui seberapa aktif seseorang melakukan aktivitas gerakan pada saat aktivitas di depan komputer/laptop. Untuk itu di lakukan penyebaran kuesioner untuk mengambil kesimpulan seberapa aktif seseorang bergerak pada saat beraktivitas di depan komputer/laptop. Data tersebut dapat dijadikan acuan pemanfaatan metode deteksi gerakan.

Kuesioner terdiri dari dua pertanyaan yaitu pertanyaan terbuka dan tertutup. Dimana, $H_0 = P_1 = P_2 = 0.5$ (Aktivitas gerakan seseorang di depan komputer/laptop adalah cenderung sering), dan $H_a = P_1 \neq P_2 \neq 0.5$.

→ NPar Tests

Descriptive Statistics								
	N	Mean	Std. Deviation	Minimum	Maximum	Percentiles		
						25th	50th (Median)	75th
CENDRUNG	30	,87	,35	0	1	1,00	1,00	1,00

Binomial Test					
	Category	N	Observed Prop.	Test Prop.	Asymp. Sig. (2-tailed)
CENDRUNG	Group 1	1	,87	,50	,000 ^a
	Group 2	0	,13		
	Total	30	1,00		

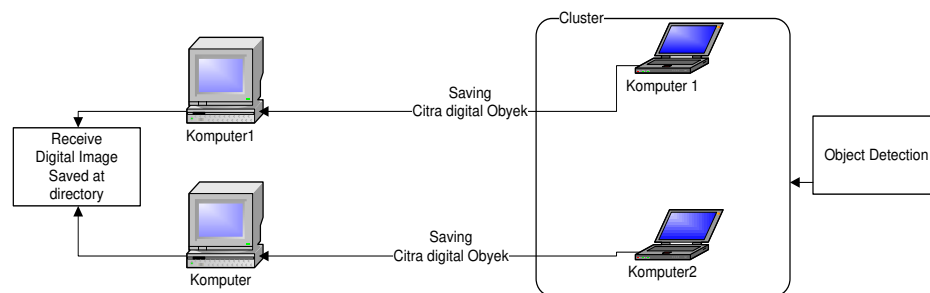
a. Based on Z Approximation.

Gambar 1 Hasil Kuesioner Dengan Binomial Test

Hasil kuesioner yang tertera pada gambar 1, nilai sigma lebih dari 0.05 maka kondisi yang diterima adalah H_0 . Analisa kesimpulan menunjukkan bahwa Aktivitas gerakan seseorang di depan komputer adalah cenderung

3. Perancangan Infrastruktur Sistem

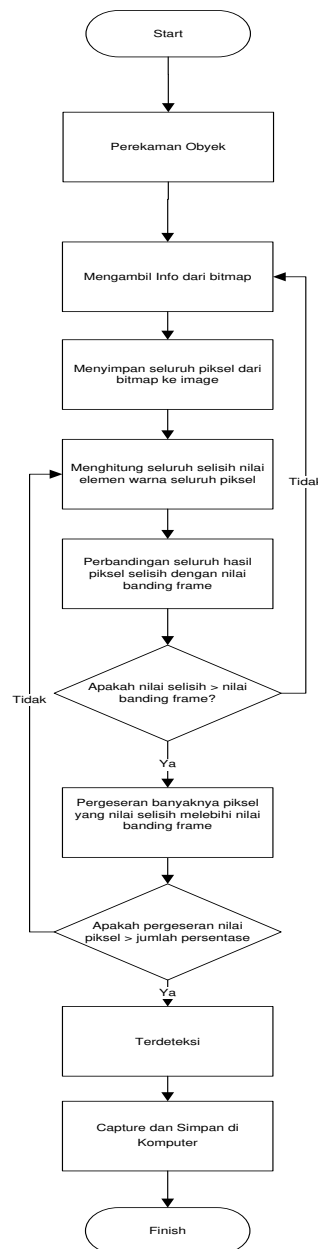
Pada gambar 2 perancangan alur program memberikan gambaran secara visual melalui sebuah gambar untuk dapat memahami bagaimana sistem bekerja dalam pengambilan citra *digital* obyek dengan pendeteksian gerakan yang dilakukan obyek. Sistem berkolaborasi bersama melakukan fungsinya untuk mendeteksi obyek.



Gambar 2 Infrastruktur Sistem

Pada gambar 2, dapat dilihat komputer1 dan komputer2 melakukan proses *pengambilan citra digital obyek* dengan menggunakan Obtion *Remote* Versi 1.0. Ketika sistem Obtion *Remote* telah aktif berjalan, akan melakukan proses perekaman obyek dan *capture*-an obyek dengan pendeteksian adanya gerakan pada obyek tersebut. Setelah proses pendeteksian gerakan obyek terdeteksi, maka akan dilakukan eksekusi pengambilan gambar obyek yang berupa citra *digital* melalui proses *capture webcam* dan selanjutnya melakukan penyimpanan hasil tangkapan obyek yang terdeteksi berupa citra *digital* ke komputer dimana sistem *obtion* berjalan. Komputer akan menyimpan hasil deteksi obyek berupa citra *digital* pada direktori khusus yang ditunjuk di

komputer tersebut. Proses deteksi obyek dilakukan secara otomatis selama sistem deteksi obyek di komputer aktif.



Gambar 3 Perancangan *Flowchart* Pengambilan *Object Detection*

Algoritma

Algoritma merupakan konsep dasar dari sebuah program. Sedangkan dalam pembuatan algoritma diperlukan daya nalar yang baik dan logis. Dengan pertimbangan, sebuah algoritma terdiri dari langkah-langkah penyelesaian masalah. Langkah-langkah algoritma dibatasi oleh obyek masalah yang dihadapi. Pada gambar 3, merupakan algoritma proses pendeteksian obyek

dimana dalam obyek masalah yang dihadapi adalah bagaimana melakukan peng-*capture*-an dengan obyek pendeteksinya adalah gerakan obyek yang dideteksi tersebut

Pseucode

Pseucode perancangan *Obtion Remote* adalah sebagai berikut :

1. Melakukan pengaktifan *preview*

```
VideoCap1.DriverIndex = 0;
VideoCap1.DriverOpen  = true;
VideoCap1.VideoPreview = true;
```

2. Melakukan perekaman obyek *active front object* dan pengambilan info *bitmap*

```
info = VideoCap1.BitMapInfo;
lebar = info.bmiHeader.biWidth;
tinggi = info.bmiHeader.biHeight;
FrameToBitmap(Image1.Picture.Bitmap, lpvhdr^.lpData, info);
FrameToBitmap(Image2.Picture.Bitmap, lpvhdr^.lpData, info);
```

3. Melakukan perhitungan dan perbandingan nilai banding *frame*

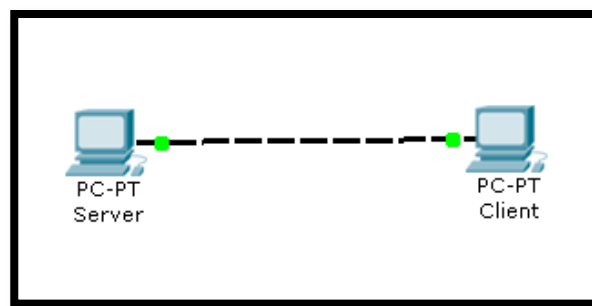
```
for y = 0 to tinggi-1 do
for x = 0 to lebar-1 do
BlueAsl[x,y] = P[3*x];
GreenAsl[x,y] = P[3*x+1];
RedAsl[x,y] = P[3*x+2];
bandingframe = UpDownBanding.Position;
geser = 0;
for y = 0 to tinggi-1 do
for x = 0 to lebar-1 do
selisih = (Abs(RedAsl[x,y]- RPalet[x,y])+Abs(GreenAsl[x,y]-
GPalet[x,y])+ Abs(BlueAsl[x,y]-BPalet[x,y])) / 3;
if selisih > bandingframe then
RedHsl[x,y] = 0;
GreenHsl[x,y] = 0;
BlueHsl[x,y] = 0;
geser = geser+1
else
RedHsl[x,y] = 255;
GreenHsl[x,y] = 255;
BlueHsl[x,y] = 255;
```

4. Melakukan perhitungan pergeseran dan perbandingan nilai persentase

```
for y = 0 to tinggi-1 do
for x = 0 to lebar-1 do
P[3*x] = BlueHsl[x,y];
P[3*x+1] = GreenHsl[x,y];
P[3*x+2] = RedHsl[x,y];
RPalet[x,y] = RedAsl[x,y];
GPalet[x,y] = GreenAsl[x,y];
BPalet[x,y] = BlueAsl[x,y];
geser > persentase/100*lebar*tinggi
```

4. Pengujian dan Analisa

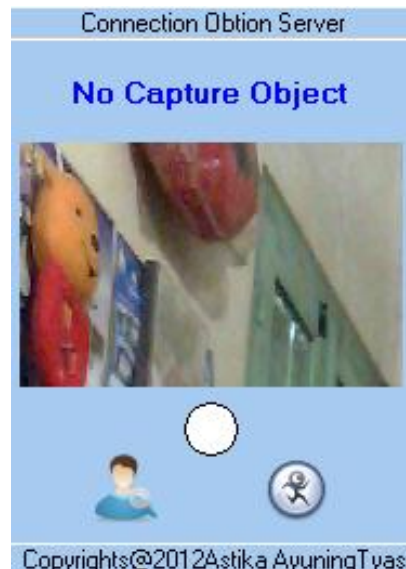
Pengujian aplikasi *Obtion Remote* Versi 1.0 dilakukan pada jaringan *localhost*, jaringan lokal (LAN Kabel dan *WiFi*), dan jaringan internet. *Obtion Remote* Versi 1.0 menggunakan dua komputer. Langkah pertama, mempersiapkan alat-alat yang diperlukan (yang telah disebutkan diatas) dan setelah terpasang pada komputer yang akan diujicobakan, selanjutnya proses instalasi *Obtion Remote* Versi 1.0 pada komputer *server* dan pada komputer yang bertindak sebagai *client* (lihat gambar 4.1 sampai gambar 4.3). Aplikasi di komputer *server* hanya berfungsi sebagai media pengujian pengiriman hasil pengambilan citra *digital* obyek yang berhasil terdeteksi dan tersimpan otomatis di *client*. Pengujian pengiriman hasil tersebut dilakukan pada beberapa infrastruktur jaringan (*localhost*, LAN Kabel, *WiFi*, internet). Sedangkan komputer *client* digunakan untuk menguji implementasi dari metode pengambilan obyek menggunakan deteksi gerakan obyek.



Gambar 4 Perancangan Simulasi Uji Coba

Uji Coba Deteksi Obyek

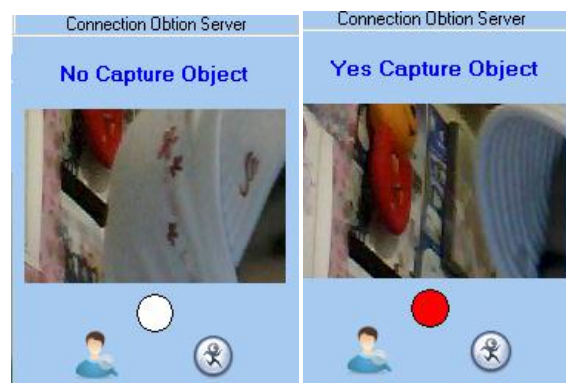
Pada *Obtion Remote* Versi 1.0 tampilan awalnya berupa *Form Obtion* (lihat gambar 5). *Form* utama yang melakukan proses pengambilan citra *digital* obyek melalui deteksi gerakan yang dilakukan obyek tersebut selama proses perekaman obyek. Sistem *Obtion* melakukan pengambilan obyek otomatis dengan metode pengambilan berdasarkan gerakan obyek. Gerakan obyek yang dideteksi adalah gerakan obyek terhadap dirinya sendiri. Setelah gerakan obyek berhasil terdeteksi, akan dilakukan pengambilan obyek berupa citra *digital* melalui hasil perekaman obyek pada *webcam*. Hasil dari pengambilan obyek akan langsung tersimpan pada komputer yang menjalankan sistem *Obtion* tersebut.



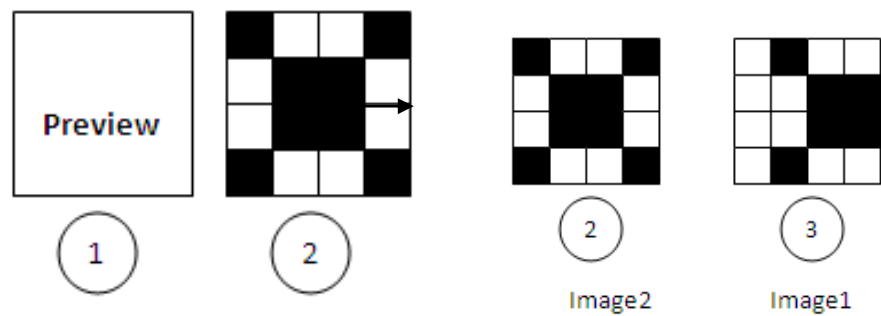
Gambar 5 *Form Obtion*

Pada gambar 5, tampilan ketika awal dilakukannya proses pengambilan citra *digital* obyek, kondisi lampu indikator masih berwarna putih (tidak ada obyek yang terdeteksi) dan informasi label “*No Capture Object*” (belum dilakukan eksekusi pengambilan obyek).

Selama proses perekaman obyek, apabila terdeteksi adanya gerakan obyek akan dilakukan pengambilan obyek berupa citra *digital* melalui proses peng-*capture*-an dengan *webcam*. Proses pendeteksian obyek dapat terlihat pada gambar 6, dimana kondisi awal informasi belum adanya obyek yang terdeteksi. Setelah terdeteksi gerakan yang dilakukan obyek, kondisi lampu indikator berwarna merah (obyek terdeteksi) dan informasi label “*Yes Capture Object*” (dilakukan eksekusi pengambilan obyek dan menyimpannya langsung pada komputer yang menjalankan sistem *Obtion*). Pada gambar 7 terdapat proses pendeteksian dari gerakan yang dilakukan obyek. Dari hasil perekaman obyek (*preview*) sampai transformasi informasi *bitmap* dalam *buffer* kamera yang tertampung dalam *frame* terhadap seluruh total piksel yang telah mengalami proses modifikasi perhitungan. Gerakan obyek akan terbaca apabila ada perubahan nilai letak piksel keseluruhan yang ada dalam *frame*. Hasil dari pengambilan citra *digital* obyek dapat dilihat pada gambar 8. Ukuran citra *digital* yang dihasilkan setiap proses pengambilan obyek adalah sama (1.17 MB). Ukuran tersebut dipengaruhi oleh *template frame* yang digunakan pada komponen *JHVideoCap* dalam format matriks 640x480.



Gambar 6 Proses Deteksi dan Pengambilan Obyek



Gambar 7 Proses Deteksi Gerakan Obyek



Gambar 8 Hasil Deteksi dan Pengambilan Obyek

Uji Coba Obyek

Pada uji coba obyek dilakukan beberapa pengambilan macam obyek. Uji coba bermacam obyek ini dilakukan untuk mengetahui keakuratan metode pembacaan gerakan pada obyek yang sedang dideteksi dan akan dilakukan proses pengambilan citra *digital* apabila gerakan yang dilakukan obyek tersebut terdeteksi. Hasil dari pengujian bermacam gerakan obyek ini dapat dilihat pada tabel 1.

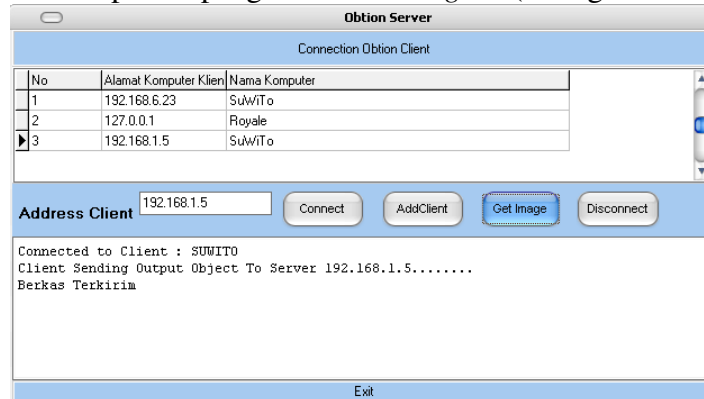
Tabel 1 Hasil Uji Coba Obyek

No	Obyek	Keterangan Gerakan	Hasil	Waktu	IP/Hostname
1.	Manusia	wajah menghadap <i>webcam</i> , diam saja	<i>Not Detected</i>	14:02:22-14:02:29	192.168.1.8 Royale
2.	Manusia	wajah menghadap ke samping, dan bergerak geleng sedikit	<i>Detected</i>	14:08:08-14:08:12	192.168.1.8 Royale
3.	Boneka Beruang	Obyek boneka di depan <i>webcam</i> , digerak-gerakan bonekanya	<i>Detected</i>	14:09:20-14:09:28	192.168.1.8 Royale
4.	Boneka Beruang	Obyek boneka di depan <i>webcam</i> , tanpa digerak-gerakan bonekanya	<i>Not Detected</i>	14:10:46-14:10:53	192.168.1.8 Royale
5.	Boneka Tupai	Di depan <i>webcam</i> dan bergerak-gerak	<i>Detected</i>	14:11:06-14:11:15	192.168.1.8 Royale
6.	Manusia dengan topeng	Di depan <i>webcam</i> , kepala bergeleng	<i>Detected</i>	14:12:54-14:12:59	192.168.1.8 Royale
7.	Manusia dengan topeng	Di depan <i>webcam</i> dan menutupi <i>webcam</i>	<i>Not Detected</i>	14:17:11-14:18:11	192.168.1.8 Royale

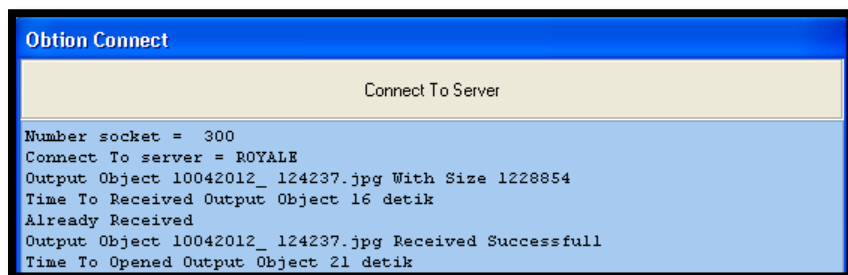
8.	Hewan Kucing	Di depan <i>webcam</i> dan bergerak-gerak	<i>Detected</i>	14:18:20-14:18:28	192.168.1.8 Royale
9.	Hewan Ayam	Di depan <i>webcam</i> dan bergerak-gerak	<i>Detected</i>	14:19:30-14:19:35	192.168.1.8 Royale
10.	Hewan Kelinci	Di depan <i>webcam</i> dan bergerak-gerak	<i>Detected</i>	14:20:05-14:20:08	192.168.1.8 Royale

Uji Coba Jaringan

Pengujian jaringan digunakan untuk menguji proses pengiriman hasil deteksi obyek berupa citra *digital* obyek. Dimana dalam proses pengujian menggunakan salah satu bagian fungsi aplikasi Obtion *Remote* Versi 1.0 yang dijalankan secara terpisah dengan proses pendeteksian dan pengambilan obyek. Sistem pengiriman tersebut dijalankan secara terpisah dan berada pada komputer yang bertindak sebagai *server* (lihat gambar 9)., sedangkan *client* (sistem yang melakukan proses pendeteksian) akan memberikan tampilan informasi komunikasi dengan *server* beserta keterangan dalam proses pengiriman citra *digital* (lihat gambar 10).



Gambar 9 Implementasi Sistem Pengujian Jaringan Di *Server*



Gambar 10 Implementasi Sistem Pengujian Jaringan Di *Client*

Uji Coba Jaringan *Localhost*

Pada uji coba jaringan *localhost*, hasil dari proses pengambilan citra *digital* obyek dengan deteksi gerakan yang dilakukan obyek akan dikirim ke *server* pada jaringan *localhost*. Pada tabel 2 adalah hasil dari uji coba pengiriman citra *digital* yang dilakukan pada jaringan *localhost*.

Tabel 2 Hasil Uji Coba Jaringan *Localhost*

No.	File Image	Size Image	File Received (Seconds)	File Open (Seconds)	Interval Seconds (Received-Open)
1.	10042012_124232.jpg	1228854(1.17MB)	8	10	2
2.	10042012_124234.jpg	1228854(1.17MB)	6	8	2

3.	10042012_124245.jpg	1228854(1.17MB)	9	11	2
4.	10042012_124247.jpg	1228854(1.17MB)	8	10	2
5.	10042012_124256.jpg	1228854(1.17MB)	12	14	2
6.	10042012_124301.jpg	1228854(1.17MB)	7	9	2
7.	10042012_124310.jpg	1228854(1.17MB)	7	9	2
8.	10042012_130121.jpg	1228854(1.17MB)	6	8	2
9.	10042012_130127.jpg	1228854(1.17MB)	6	9	3
10.	10042012_130130.jpg	1228854(1.17MB)	6	8	2
11.	10042012_130133.jpg	1228854(1.17MB)	6	8	2
12.	10042012_130142.jpg	1228854(1.17MB)	6	8	2
13.	10042012_130146.jpg	1228854(1.17MB)	7	9	2
14.	10042012_130216.jpg	1228854(1.17MB)	7	9	2
15.	10042012_130223.jpg	1228854(1.17MB)	7	9	2

Uji Coba Jaringan LAN (Kabel)

Pada uji coba jaringan LAN dengan kabel, hasil dari proses pengambilan citra *digital* obyek dengan deteksi gerakan yang dilakukan obyek akan dikirim ke *server* pada jaringan LAN (kabel). Pada tabel 3 adalah hasil dari uji coba pengiriman citra *digital* yang dilakukan pada jaringan LAN (kabel).

Tabel 3 Hasil Uji Coba Jaringan LAN (Kabel)

No.	File Image	Size Image	File Received (Seconds)	File Open (Seconds)	Interval Seconds (Received-Open)
1.	10042012_124232.jpg	1228854(1.17MB)	13	14	1
2.	10042012_124234.jpg	1228854(1.17MB)	15	17	2
3.	10042012_124245.jpg	1228854(1.17MB)	8	10	2
4.	10042012_124247.jpg	1228854(1.17MB)	12	13	1
5.	10042012_124256.jpg	1228854(1.17MB)	10	13	3
6.	10042012_124301.jpg	1228854(1.17MB)	8	10	2
7.	10042012_124310.jpg	1228854(1.17MB)	7	9	2
8.	10042012_130121.jpg	1228854(1.17MB)	8	9	1
9.	10042012_130127.jpg	1228854(1.17MB)	8	10	2
10.	10042012_130130.jpg	1228854(1.17MB)	8	10	2
11.	10042012_130133.jpg	1228854(1.17MB)	14	16	2
12.	10042012_130142.jpg	1228854(1.17MB)	9	11	2
13.	10042012_130146.jpg	1228854(1.17MB)	8	11	3
14.	10042012_130216.jpg	1228854(1.17MB)	7	10	3
15.	10042012_130223.jpg	1228854(1.17MB)	10	12	2

Uji Coba Jaringan LAN (WiFi)

Pada uji coba jaringan LAN (WiFi), hasil dari proses pengambilan citra *digital* obyek dengan deteksi gerakan yang dilakukan obyek akan dikirim ke *server* pada jaringan LAN (WiFi). Pada tabel 4 adalah hasil dari uji coba pengiriman citra *digital* yang dilakukan pada jaringan LAN (WiFi).

Tabel 4 Hasil Uji Coba Jaringan LAN (WiFi)

No.	File Image	Size Image	File Received (Seconds)	File Open (Seconds)	Interval Seconds (Received-Open)
1.	10042012_124232.jpg	1228854(1.17MB)	15	18	3
2.	10042012_124234.jpg	1228854(1.17MB)	19	24	5
3.	10042012_124245.jpg	1228854(1.17MB)	25	27	2
4.	10042012_124247.jpg	1228854(1.17MB)	29	30	1
5.	10042012_124256.jpg	1228854(1.17MB)	31	32	1

6.	10042012_124301.jpg	1228854(1.17MB)	33	34	1
7.	10042012_124310.jpg	1228854(1.17MB)	30	31	1
8.	10042012_130121.jpg	1228854(1.17MB)	28	30	2
9.	10042012_130127.jpg	1228854(1.17MB)	26	28	2
10.	10042012_130130.jpg	1228854(1.17MB)	25	27	2
11.	10042012_130133.jpg	1228854(1.17MB)	30	32	2
12.	10042012_130142.jpg	1228854(1.17MB)	29	31	2
13.	10042012_130146.jpg	1228854(1.17MB)	28	29	1
14.	10042012_130216.jpg	1228854(1.17MB)	25	26	1
15.	10042012_130223.jpg	1228854(1.17MB)	27	28	1

Uji Coba Jaringan Internet

Pada uji coba jaringan internet, hasil dari proses pengambilan citra *digital* obyek dengan deteksi gerakan yang dilakukan obyek akan dikirim ke *server* pada jaringan internet. Pada tabel 5 adalah hasil dari uji coba pengiriman citra *digital* yang dilakukan pada jaringan internet.

Tabel 5 Hasil Uji Coba Jaringan Internet

No.	File Image	Size Image	File Received (Seconds)	File Open (Seconds)	Interval Seconds (Received-Open)
1.	10042012_124232.jpg	1228854(1.17MB)	19	23	4
2.	10042012_124234.jpg	1228854(1.17MB)	28	32	4
3.	10042012_124245.jpg	1228854(1.17MB)	39	41	2
4.	10042012_124247.jpg	1228854(1.17MB)	48	52	4
5.	10042012_124256.jpg	1228854(1.17MB)	57	60	3
6.	10042012_124301.jpg	1228854(1.17MB)	65	67	2
7.	10042012_124310.jpg	1228854(1.17MB)	73	76	3
8.	10042012_130121.jpg	1228854(1.17MB)	81	84	3
9.	10042012_130127.jpg	1228854(1.17MB)	89	91	2
10.	10042012_130130.jpg	1228854(1.17MB)	97	100	3
11.	10042012_130133.jpg	1228854(1.17MB)	104	106	2
12.	10042012_130142.jpg	1228854(1.17MB)	112	114	2
13.	10042012_130146.jpg	1228854(1.17MB)	120	124	2
14.	10042012_130216.jpg	1228854(1.17MB)	130	133	3
15.	10042012_130223.jpg	1228854(1.17MB)	138	142	4

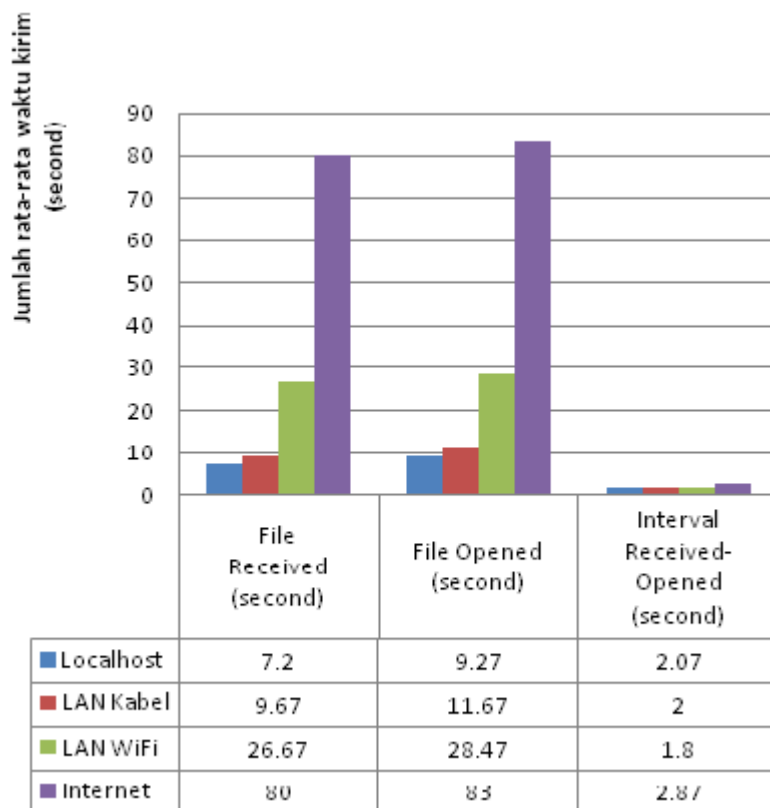
Analisa Pendeteksian Obyek Dan Pengambilan Citra *Digital* Obyek

Analisa yang dilakukan pada aplikasi *Obtion Remote* Versi 1.0 pada penerapan metode pengambilan obyek (citra *digital*) dengan deteksi gerakan obyek terhadap dirinya sendiri. Mendeteksi adanya gerakan obyek adalah dengan membandingkan setiap piksel dalam sebuah *frame* dengan piksel pada posisi yang sama dalam *frame* sebelumnya (lihat gambar 6). Apabila tidak ada obyek yang bergerak dalam *frame* tersebut, maka hasil perbandingan akan memberikan hasil yang relatif kecil.

Analisa Pendeteksian Berbagai Gerak Obyek

Mendeteksi dengan berbagai obyek tidak mempengaruhi proses pendeteksian jenis obyek yang bergerak. Semua obyek dapat terdeteksi jika melakukan gerakan selama proses perekaman obyek dengan *webcam*. Hal tersebut dikarenakan metode yang digunakan adalah perbandingan setiap piksel dalam sebuah *frame* dengan piksel pada posisi yang sama dalam *frame* sebelumnya. Spesifik obyek yang tertangkap dalam proses perekaman tidak menjadi bahan perbandingan. Karena info yang diambil dalam *buffer* kamera hanyalah info tempat keseluruhan letak piksel terhadap *frame*. Dimana sebelumnya dilakukan perhitungan banyaknya piksel yang nilai selisih jumlahnya melebihi nilai bidang *frame* dengan pengatur perubahan letak *frame*

terhadap gerakan obyek. Sehingga apapun obyek yang direkam pada *webcam* melakukan aktivitas (gerakan) akan dapat terdeteksi.



Gambar 11 Hasil Pengiriman Citra *Digital* Obyek di Jaringan

Pada gambar 11, semakin ke kanan grafik menunjukkan waktu yang dibutuhkan semakin banyak. Untuk area *localhost* dibutuhkan waktu paling sedikit karena transfer data masih pada satu komputer saja (hanya berpindah direktori). Waktu transfer data pada jaringan LAN kabel dan LAN *WiFi* menghasilkan perbedaan yang cukup signifikan, hal ini dipengaruhi oleh media dalam jaringan yang digunakan. Waktu transfer paling lama ditunjukkan pada hasil uji coba yang dilakukan pada jaringan internet, beberapa faktor seperti trafik kecepatan, jumlah kuota akses, dll mempengaruhi proses pengiriman.

5. Kesimpulan

Dalam melakukan pembuatan dan pengujian aplikasi *Obtion Remote* Versi 1.0 ini terdapat beberapa kesimpulan yang dapat diambil, diantaranya adalah :

1. Aplikasi *Obtion Remote* Versi 1.0 digunakan untuk pengambilan obyek berupa citra *digital* dengan menerapkan metode deteksi gerakan yang dilakukan obyek.
2. Metode deteksi gerakan obyek mendeteksi berbagai obyek apapun yang melakukan gerakan di depan *webcam* yang sedang dilakukan proses perekaman dan pendeteksian secara otomatis selama sistem aktif berjalan.
3. Hasil dari pendeteksian obyek berupa citra *digital* obyek langsung tersimpan pada komputer *client* yang menjalankan sistem *obtion* dan dikirimkan ke *server* dengan waktu transfornya bergantung dengan media jaringan yang diujikan.

4. Kelemahan aplikasi *Obtion Remote* Versi 1.0 adalah ketergantungan terhadap komponen yang digunakan yaitu *JHVideoCap* sehingga mempengaruhi dalam proses pendeteksian obyek dan pengambilan citra *digital* obyek karena terdapat fungsi-fungsi komponen yang bersifat *read only*.

Saran

Adapun saran yang diberikan pada penelitian ini adalah sebagai berikut :

1. Dalam pengembangannya ke depan, metode deteksi gerakan obyek dalam pengambilan citra *digital* obyek dapat dispesifikkan pada jenis obyek tertentu dan gerakan bagian obyek tertentu (seperti *face detection*).
2. *Obtion Remote* Versi 1.0 dengan menerapkan deteksi gerakan obyek dapat menjadi ide awal untuk mengembangkan aplikasi dibidang forensik sebagai solusi untuk IT forensik mendapatkan bukti pengambilan obyek tertentu melalui kamera.

6. Daftar Pustaka

Achmad, Balza, 2011, *Pemrograman Delphi untuk Aplikasi Mesin Visi menggunakan Webcam*, Gava Media, Yogyakarta.

Fadlisyah, Rizal, 2011, *Pemrograman Computer Vision pada Video menggunakan Delphi+VisionLab VCL 4.0.1*, Graha Ilmu, Yogyakarta.

Suarna, Nana, 2007, *Pengantar LAN (Local Area Network)*, Yrama Widya, Bandung.

Agushinta R D 2008, *perbandingan kinerja metode deteksi tepi.pdf*,
http://repository.gunadarma.ac.id:8000/1851/1/UGArtikelDewiAR_edit.pdf, diakses
tanggal 27 Desember 2011

Shinta, dkk 2009, *aplikasi webcam untuk mendeteksi gerakan suatu objek.pdf*,
http://eprints.undip.ac.id/25283/1/2_Makalah_L2F002577.pdf,
diakses tanggal 27 Desember 2011

Prayogi S 2009, *sistem deteksi wajah.pdf*,
<http://repo.eepis-its.edu/474/1/854.pdf>, diakses tanggal 27 Desember 2011.